


# Date

 Apr 29, 2020 09:12

## Supported Clients

SmartClient WebClient NGClient MobileClient

## Methods Summary

Number	UTC(year, month)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	UTC(year, month, date)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	UTC(year, month, date, hrs)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	UTC(year, month, date, hrs, min)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	UTC(year, month, date, hrs, min, sec)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	UTC(year, month, date, hrs, min, sec, ms)	Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.
Number	getDate()	Gets the day of month in local time
Number	getDay()	Gets the day of the week (sunday = 0) in local time
Number	getFullYear()	Gets the full year of the date in local time
Number	getHours()	Gets the hours of the date in local time
Number	getMilliseconds()	Gets the milliseconds of the date in local time
Number	getMinutes()	Gets the minutes of the date in local time
Number	getMonth()	Gets the month of the date in local time
Number	getSeconds()	Gets the seconds of the date in local time
Number	getTime()	The value returned by the getTime method is the number of milliseconds since 1 January 1970 00:00:00.
Number	getTimezoneOffset()	Gets the number of minutes between GMT and this date.
Number	getUTCDate()	Gets the UTC date.
Number	getUTCDay()	Gets the day in UTC time.
Number	getUTCFullYear()	Gets the full year in UTC time.
Number	getUTCHours()	Gets the hours in UTC time.
Number	getUTCMilliseconds()	Gets the milliseconds in UTC time.
Number	getUTCMinutes()	Gets the minutes in UTC time.
Number	getUTCMonth()	Gets the month in UTC time.
Number	getUTCSeconds()	Gets the seconds in UTC time.
Number	now()	Returns the milliseconds elapsed since 1 January 1970 00:00:00 UTC up until now.
Number	parse(s)	Takes a date string (such as "Dec 25, 1995") and returns the number of milliseconds since January 1, 1970, 00:00:00 UTC.
void	setDate(dayValue)	Sets the date.
void	setFullYear(yearValue)	Sets the full year of the date.
void	setFullYear(yearValue, monthValue, dayValue)	Sets the full year of the date.
void	setHours(hoursValue)	Sets the hours of the date.
void	setHours(hoursValue, minutesValue)	Sets the hours of the date.
void	setHours(hoursValue, minutesValue, secondsValue)	Sets the hours of the date.
void	setHours(hoursValue, minutesValue, secondsValue, msValue)	Sets the hours of the date.
void	setMilliseconds(millisecondsValue)	Sets the milliseconds of the date.
void	setMinutes(minutesValue)	Sets the minutes of the date.
void	setMinutes(minutesValue, secondsValue)	Sets the minutes of the date.
void	setMinutes(minutesValue, secondsValue, msValue)	Sets the minutes of the date.
void	setMonth(monthValue)	Sets the month of the date.
void	setMonth(monthValue, dayValue)	Sets the month of the date.
void	setSeconds(secondsValue)	Sets the seconds of the date.
void	setSeconds(secondsValue, msValue)	Sets the seconds of the date.
void	setTime(timeValue)	Sets the milliseconds of the date.
void	setUTCDate(dayValue)	Sets the UTC date.
void	setUTCFullYear(yearValue)	Sets the year in UTC time.
void	setUTCFullYear(yearValue, monthValue)	Sets the year in UTC time.
void	setUTCFullYear(yearValue, monthValue, dayValue)	Sets the year in UTC time.
void	setUTCHours(hoursValue)	Sets the hours in UTC time.

void	<a href="#">setUTCHours(hoursValue, minutesValue)</a>	Sets the hours in UTC time.
void	<a href="#">setUTCHours(hoursValue, minutesValue, secondsValue)</a>	Sets the hours in UTC time.
void	<a href="#">setUTCHours(hoursValue, minutesValue, secondsValue, msValue)</a>	Sets the hours in UTC time.
void	<a href="#">setUTCMilliseconds(millisecondsValue)</a>	Sets the milliseconds in UTC time.
void	<a href="#">setUTCMinutes(minutesValue)</a>	Sets the minutes in UTC time.
void	<a href="#">setUTCMinutes(minutesValue, secondsValue)</a>	Sets the minutes in UTC time.
void	<a href="#">setUTCMinutes(minutesValue, secondsValue, msValue)</a>	Sets the minutes in UTC time.
void	<a href="#">setUTCMonth(monthValue)</a>	Sets the month in UTC time.
void	<a href="#">setUTCMonth(monthValue, dayValue)</a>	Sets the month in UTC time.
void	<a href="#">setUTCSeconds(secondsValue)</a>	Sets the seconds in UTC time.
void	<a href="#">setUTCSeconds(secondsValue, msValue)</a>	Sets the seconds in UTC time.
String	<a href="#">toDateString()</a>	Returns a string version of the date.
String	<a href="#">toISOString()</a>	Returns a string version of the UTC value of the date.
String	<a href="#">toLocaleDateString()</a>	Returns a string version of the local time zone of the date.
String	<a href="#">toLocaleString()</a>	Returns a string version of the local time zone of the date.
String	<a href="#">toLocaleTimeString()</a>	Returns a string version of the local time zone of the date.
String	<a href="#">toTimeString()</a>	Returns a string version of the date.
String	<a href="#">toUTCString()</a>	Returns a string version of the UTC value of the date.
Number	<a href="#">valueOf()</a>	Return integer milliseconds count

## Methods Details

### UTC(year, month)

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

#### Parameters

[Number](#) year A year after 1900.

[Number](#) month A number between 0 and 11.

#### Returns

[Number](#)

#### Supported Clients

SmartClient,WebClient,NGClient,MobileClient

#### Sample

```
// The number of milliseconds in the first minute after 1970 January 1st.
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

### UTC(year, month, date)

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

#### Parameters

[Number](#) year A year after 1900.

[Number](#) month A number between 0 and 11.

[Number](#) date A number between 1 and 31.

#### Returns

[Number](#)

#### Supported Clients

SmartClient,WebClient,NGClient,MobileClient

#### Sample

```
// The number of milliseconds in the first minute after 1970 January 1st.
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

### UTC(year, month, date, hrs)

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

**Parameters**

**Number** year A year after 1900.  
**Number** month A number between 0 and 11.  
**Number** date A number between 1 and 31.  
**Number** hrs A number between 0 and 23.

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
// The number of milliseconds in the first minute after 1970 January 1st.
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

**UTC(year, month, date, hrs, min)**

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

**Parameters**

**Number** year A year after 1900.  
**Number** month A number between 0 and 11.  
**Number** date A number between 1 and 31.  
**Number** hrs A number between 0 and 23.  
**Number** min A number between 0 and 59.

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
// The number of milliseconds in the first minute after 1970 January 1st.
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

**UTC(year, month, date, hrs, min, sec)**

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

**Parameters**

**Number** year A year after 1900.  
**Number** month A number between 0 and 11.  
**Number** date A number between 1 and 31.  
**Number** hrs A number between 0 and 23.  
**Number** min A number between 0 and 59.  
**Number** sec A number between 0 and 59.

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
// The number of milliseconds in the first minute after 1970 January 1st.
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

**UTC(year, month, date, hrs, min, sec, ms)**

Takes comma-delimited date parameters and returns the number of milliseconds between January 1, 1970, 00:00:00, universal time and the specified time.

---

**Parameters**

**Number** year A year after 1900.  
**Number** month A number between 0 and 11.  
**Number** date A number between 1 and 31.  
**Number** hrs A number between 0 and 23.  
**Number** min A number between 0 and 59.  
**Number** sec A number between 0 and 59.  
**Number** ms A number between 0 and 999.

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
// The number of milliseconds in the first minute after 1970 January 1st.  
application.output(Date.UTC(1970, 00, 01, 00, 01, 00, 00)); // prints: 60000.0
```

**getDate()**

Gets the day of month in local time

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.getDate();
```

**getDay()**

Gets the day of the week (sunday = 0) in local time

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.getDay();
```

**getFullYear()**

Gets the full year of the date in local time

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.getFullYear();
```

**getHours()**

Gets the hours of the date in local time

**Returns**

**Number**

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

---

**Sample**

```
date.getHours();
```

**getMilliseconds()**

Gets the milliseconds of the date in local time

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getMilliseconds();
```

**getMinutes()**

Gets the minutes of the date in local time

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getMinutes();
```

**getMonth()**

Gets the month of the date in local time

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getMonth();
```

**getSeconds()**

Gets the seconds of the date in local time

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getSeconds();
```

**getTime()**

The value returned by the getTime method is the number of milliseconds since 1 January 1970 00:00:00.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

---

**Sample**

```
date.getTime();
```

**getTimezoneOffset()**

Gets the number of minutes between GMT and this date.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getTimezoneOffset();
```

**getUTCDate()**

Gets the UTC date.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCDate();
```

**getUTCDay()**

Gets the day in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCDay();
```

**getUTCFullYear()**

Gets the full year in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCFullYear();
```

**getUTCHours()**

Gets the hours in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

---

**Sample**

```
date.getUTCHours();
```

**getUTCMilliseconds()**

Gets the milliseconds in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCMilliseconds();
```

**getUTCMinutes()**

Gets the minutes in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCMinutes();
```

**getUTCMonth()**

Gets the month in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCMonth();
```

**getUTCSeconds()**

Gets the seconds in UTC time.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.getUTCSeconds();
```

**now()**

Returns the milliseconds elapsed since 1 January 1970 00:00:00 UTC up until now.

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

---

**Sample**

```
var timestamp = Date.now();
```

**parse(s)**

Takes a date string (such as "Dec 25, 1995") and returns the number of milliseconds since January 1, 1970, 00:00:00 UTC.

**Parameters**

[String](#) s The date string to parse

**Returns**

[Number](#)

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
var str = Date.parse("Wed, 09 Aug 1995 00:00:00 GMT");  
application.output(str);
```

**setDate(dayValue)**

Sets the date.

**Parameters**

[Number](#) dayValue;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setDate(integer);
```

**setFullYear(yearValue)**

Sets the full year of the date.

**Parameters**

[Number](#) yearValue;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setFullYear(integer);
```

**setFullYear(yearValue, monthValue)**

Sets the full year of the date.

**Parameters**

[Number](#) yearValue ;

[Number](#) monthValue;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setFullYear(integer);
```

**setFullYear(yearValue, monthValue, dayValue)**

Sets the full year of the date.



---

**Parameters**

`Number` yearValue ;  
`Number` monthValue ;  
`Number` dayValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setFullYear(integer);
```

**setHours(hoursValue)**

Sets the hours of the date.

**Parameters**

`Number` hoursValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setHours(integer);
```

**setHours(hoursValue, minutesValue)**

Sets the hours of the date.

**Parameters**

`Number` hoursValue ;  
`Number` minutesValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setHours(integer);
```

**setHours(hoursValue, minutesValue, secondsValue)**

Sets the hours of the date.

**Parameters**

`Number` hoursValue ;  
`Number` minutesValue ;  
`Number` secondsValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setHours(integer);
```

**setHours(hoursValue, minutesValue, secondsValue, msValue)**

Sets the hours of the date.

**Parameters**

`Number` hoursValue ;  
`Number` minutesValue ;  
`Number` secondsValue ;  
`Number` msValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

---

**Sample**

```
date.setHours(integer);
```

**setMilliseconds(millisecondsValue)**

Sets the milliseconds of the date.

**Parameters**

[Number](#) millisecondsValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMilliseconds(integer);
```

**setMinutes(minutesValue)**

Sets the minutes of the date.

**Parameters**

[Number](#) minutesValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMinutes(integer);
```

**setMinutes(minutesValue, secondsValue)**

Sets the minutes of the date.

**Parameters**

[Number](#) minutesValue ;

[Number](#) secondsValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMinutes(integer);
```

**setMinutes(minutesValue, secondsValue, msValue)**

Sets the minutes of the date.

**Parameters**

[Number](#) minutesValue ;

[Number](#) secondsValue ;

[Number](#) msValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMinutes(integer);
```

**setMonth(monthValue)**

Sets the month of the date.

**Parameters**

[Number](#) monthValue ;

---

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMonth(integer);
```

**setMonth(monthValue, dayValue)**

Sets the month of the date.

**Parameters**

Number monthValue ;  
Number dayValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setMonth(integer);
```

**setSeconds(secondsValue)**

Sets the seconds of the date.

**Parameters**

Number secondsValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setSeconds(integer);
```

**setSeconds(secondsValue, msValue)**

Sets the seconds of the date.

**Parameters**

Number secondsValue ;  
Number msValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setSeconds(integer);
```

**setTime(timeValue)**

Sets the milliseconds of the date.

**Parameters**

Number timeValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setTime(integer);
```

**setUTCDate(dayValue)**

Sets the UTC date.

**Parameters**

Number dayValue ;

---

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCDate(integer);
```

**setUTCFullYear(yearValue)**

Sets the year in UTC time.

**Parameters**`Number` yearValue ;**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCFullYear(integer);
```

**setUTCFullYear(yearValue, monthValue)**

Sets the year in UTC time.

**Parameters**`Number` yearValue ;`Number` monthValue ;**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCFullYear(integer);
```

**setUTCFullYear(yearValue, monthValue, dayValue)**

Sets the year in UTC time.

**Parameters**`Number` yearValue ;`Number` monthValue ;`Number` dayValue ;**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCFullYear(integer);
```

**setUTCHours(hoursValue)**

Sets the hours in UTC time.

**Parameters**`Number` hoursValue ;**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCHours(integer);
```

**setUTCHours(hoursValue, minutesValue)**

Sets the hours in UTC time.

**Parameters**

[Number](#) hoursValue ;  
[Number](#) minutesValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCHours(integer);
```

**setUTCHours(hoursValue, minutesValue, secondsValue)**

Sets the hours in UTC time.

**Parameters**

[Number](#) hoursValue ;  
[Number](#) minutesValue ;  
[Number](#) secondsValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCHours(integer);
```

**setUTCHours(hoursValue, minutesValue, secondsValue, msValue)**

Sets the hours in UTC time.

**Parameters**

[Number](#) hoursValue ;  
[Number](#) minutesValue ;  
[Number](#) secondsValue ;  
[Number](#) msValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCHours(integer);
```

**setUTCMilliseconds(millisecondsValue)**

Sets the milliseconds in UTC time.

**Parameters**

[Number](#) millisecondsValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCMilliseconds(integer);
```

**setUTCMinutes(minutesValue)**

Sets the minutes in UTC time.

**Parameters**

[Number](#) minutesValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCMinutes(integer);
```

**setUTCMinutes(minutesValue, secondsValue)**

Sets the minutes in UTC time.

**Parameters**

[Number](#) minutesValue ;

[Number](#) secondsValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setUTCMinutes(integer);
```

**setUTCMinutes(minutesValue, secondsValue, msValue)**

Sets the minutes in UTC time.

**Parameters**

[Number](#) minutesValue ;

[Number](#) secondsValue ;

[Number](#) msValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setUTCMinutes(integer);
```

**setUTCMonth(monthValue)**

Sets the month in UTC time.

**Parameters**

[Number](#) monthValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setUTCMonth(integer);
```

**setUTCMonth(monthValue, dayValue)**

Sets the month in UTC time.

**Parameters**

[Number](#) monthValue ;

[Number](#) dayValue ;

**Supported Clients**

SmartClient, WebClient, NGClient, MobileClient

**Sample**

```
date.setUTCMonth(integer);
```

**setUTCSeconds(secondsValue)**

Sets the seconds in UTC time.

**Parameters**

[Number](#) secondsValue ;

---

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCSeconds(integer);
```

**setUTCSeconds(secondsValue, msValue)**

Sets the seconds in UTC time.

**Parameters**

[Number](#) secondsValue;  
[Number](#) msValue ;

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.setUTCSeconds(integer);
```

**toDateString()**

Returns a string version of the date.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toDateString();
```

**toISOString()**

Returns a string version of the UTC value of the date.

**Returns**

[String](#) the Date object as a string in simplified extended ISO format.

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toISOString();
```

**toLocaleDateString()**

Returns a string version of the local time zone of the date.

**Returns**

[String](#)

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toLocaleDateString();
```

**toLocaleString()**

Returns a string version of the local time zone of the date.

**Returns**

[String](#)

---

**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toLocaleString();
```

**toLocaleTimeString()**

Returns a string version of the local time zone of the date.

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toLocaleTimeString();
```

**getTimeString()**

Returns a string version of the date.

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.getTimeString();
```

**toUTCString()**

Returns a string version of the UTC value of the date.

**Returns**[String](#)**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.toUTCString();
```

**valueOf()**

Return integer milliseconds count

**Returns**[Number](#)**Supported Clients**

SmartClient,WebClient,NGClient,MobileClient

**Sample**

```
date.valueOf(integer);
```